# Architecture of Scalable HPC Machines

# Topics

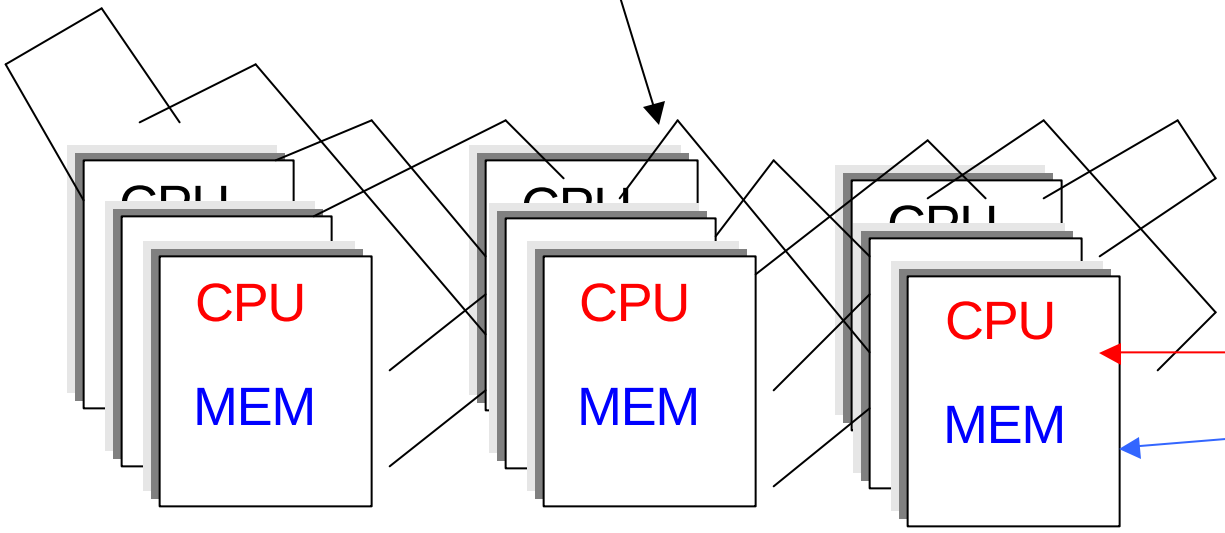
- General configuration of HPC systems
  - architecture of  each processor
  - memory and cache arrangement within each processor
  - network interconnect of each machine
- Architecture of Scalable HPC machines:
  - CRAY T3E, IBM SP, TERA MTA, Origin 2000

# Motivation

- Observed performance of most of the "real" codes is around ~ 20% of peak MFLOPS of current massively parallel machines
- It is important to optimize performance on single processor as well as the communication time to get maximum utilization of parallel codes
- Understanding of general configuration, processor architecture, memory arrangement, and network is important for optimizing both single processor and parallel performance of a code

# Basic Components of an MPP

- **Processors**
- **Memory**
- Interconnect Network

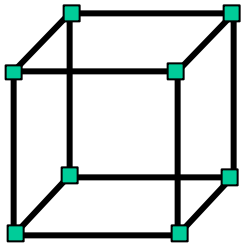


# Processor Related Terms

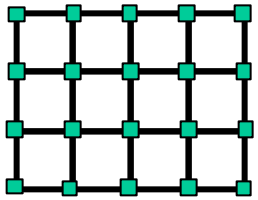
- RISC: Reduced Instruction Set Computers
- PIPELINE : Technique where multiple instructions are overlapped in execution
- SUPERSCALAR: Multiple instructions per clock period

# Network Interconnect

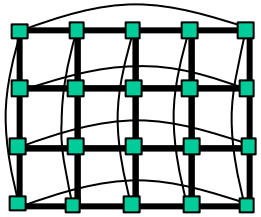
The best choice would be a fully connected network in which each processor has a direct link to every other processor. Unfortunately, this type of network would be very expensive and difficult to scale. Instead, processors are arranged in some variation of a grid, torus, or hypercube.



3-d hypercube



2-d mesh



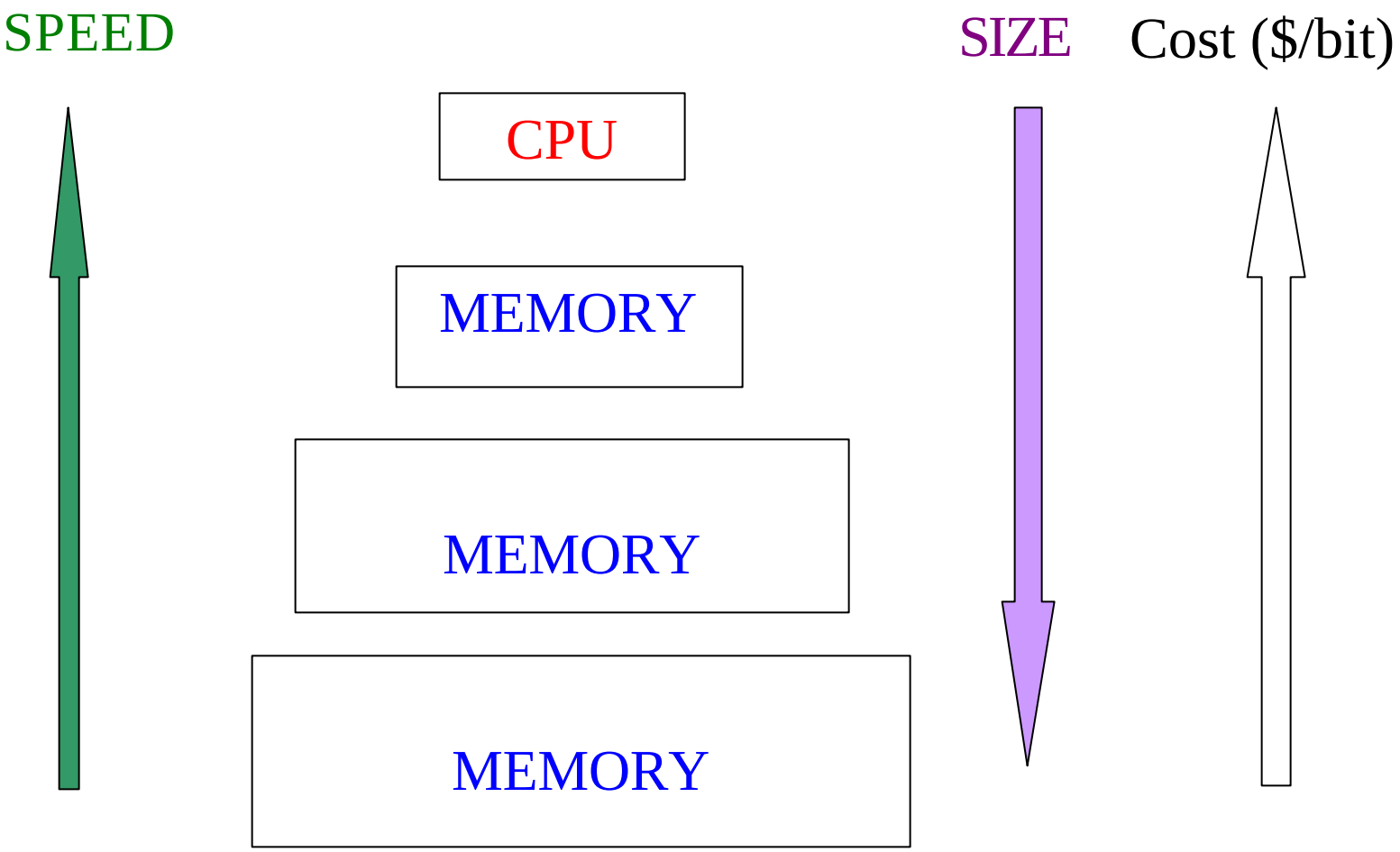
2-d torus

We won't worry too much about network topologies, since it is not the topology itself that we are interested in, but rather the affect that it has on the really important parameters: latency and bandwidth.

# Network Interconnect Related Terms

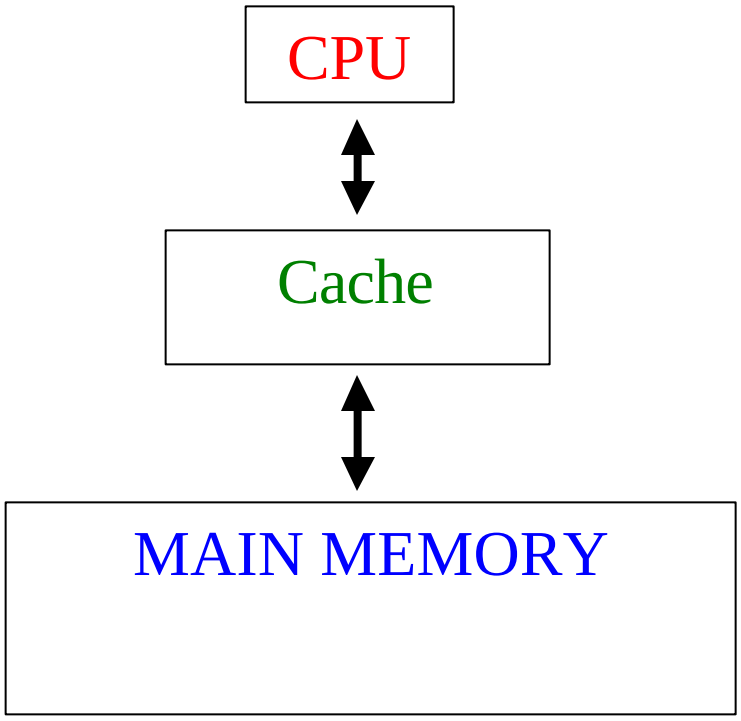
- Latency : How long does it take to start sending a "message"?
  Unit is microsecond, millisecond etc.
  (How long does it take to output results of some operations (such as floating point add, divide etc.) which are pipelined?
  Unit is Clock Period (CP).)
- Bandwidth : what data rate can be sustained once the message is started?
  Unit is bytes/sec, Mbytes/sec etc.

# Memory/Cache Related Terms



# Memory/Cache Related Terms (cont.)

Cache : Cache is the level of memory hierarchy between the CPU and main memory. Cache is much smaller than main memory and hence there is mapping of  data from main memory to cache.

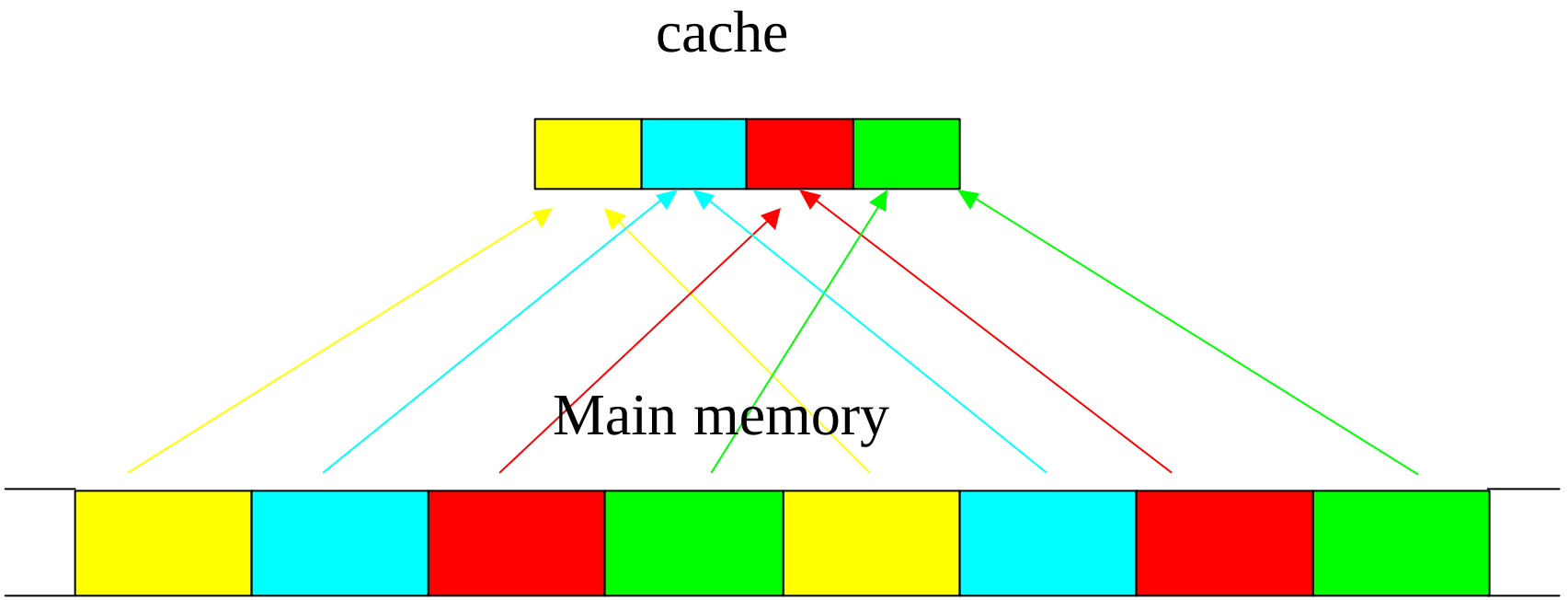


# Memory/Cache Related Terms (cont.)

- The data cache was designed with two key concepts in mind
  - Spatial Locality
    - When an element is referenced its neighbors will be referenced too
    - Cache lines are fetched together
    - Work on consecutive data elements in the same cache line
  - Temporal Locality
    - When an element is referenced, it might be referenced again soon
    - Arrange code so that date in cache is reused as often as possible

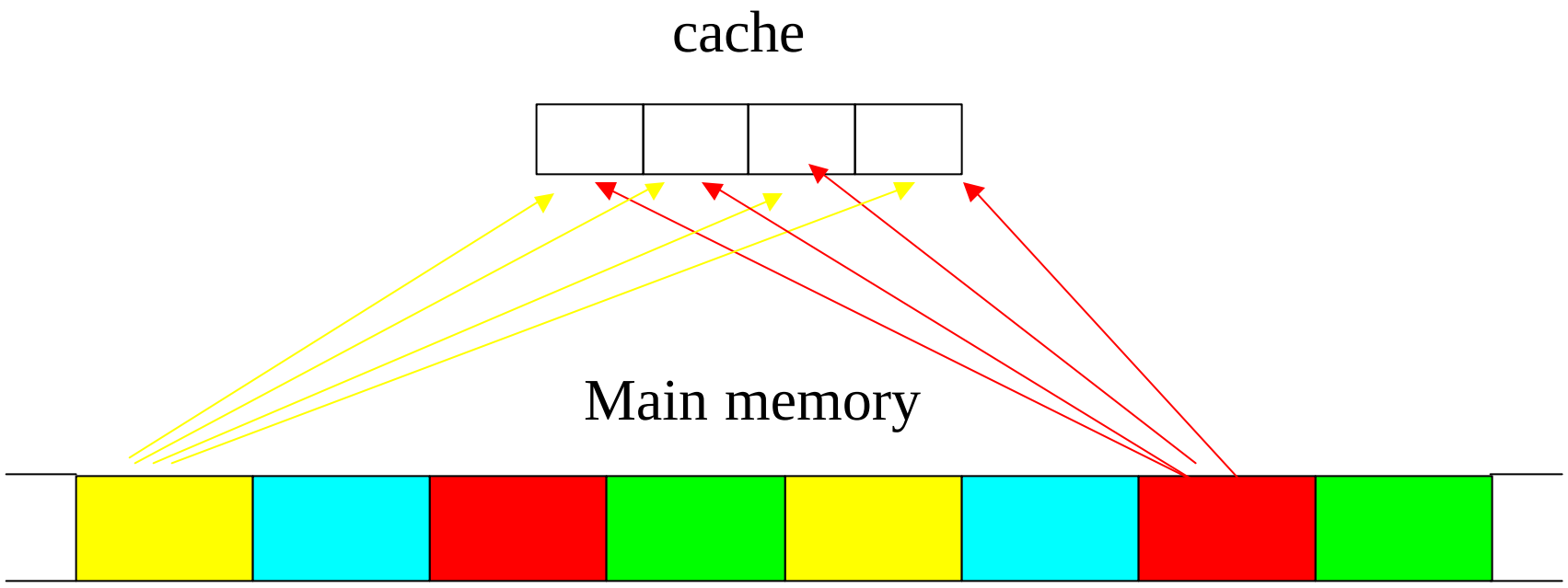
# Memory/Cache Related Terms (cont.)

Direct mapped cache: A block from main memory can go in exactly one place in the cache. This is called direct mapped because there is direct mapping from any block address in memory to a single location in the cache.



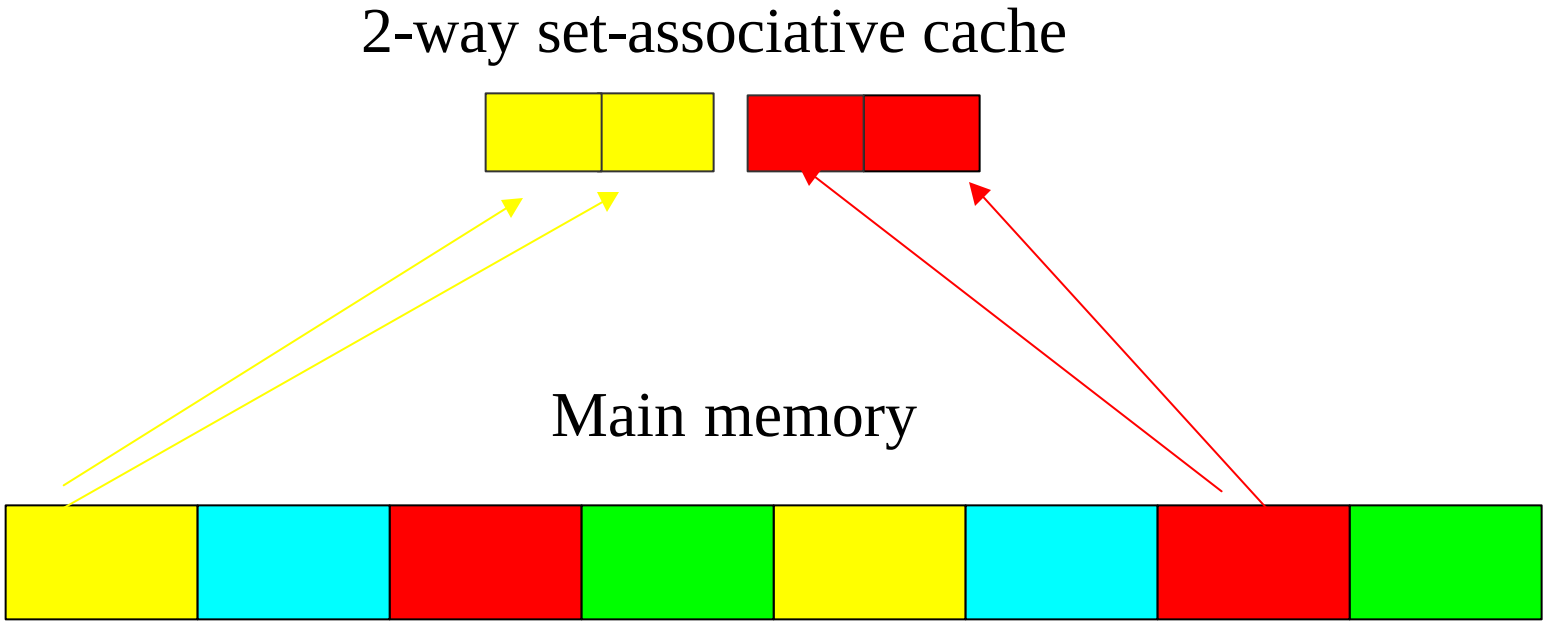
# Memory/Cache Related Terms (cont.)

Fully associative cache : A block from main memory can be placed in any location in the cache. This is called fully associative because a block in main memory may be associated with any entry in the cache.



# Memory/Cache Related Terms (cont.)

Set associative cache : The middle range of designs between direct mapped cache and fully associative cache is called set-associative cache. In a n-way set-associative cache a block from main memory can go into n (n at least 2) locations in the cache.



# Memory/Cache Related Terms (cont.)

- Least Recently Used (LRU) : Cache replacement strategy for set associative caches. The cache block that is least recently used is replaced with a new block.
- Random Replace : Cache replacement strategy for set associative caches. A cache block is randomly replaced.

# Memory/Cache Related Terms (cont.)

- ICACHE : Instruction cache
- DCACHE (L1) : Data cache closest to registers
- SCACHE (L2) : Secondary data cache
  - Data from SCACHE has to go through DCACHE to registers
  - SCACHE is larger than DCACHE
  - All processors do not have SCACHE
- TLB : Translation-lookaside buffer keeps addresses of pages ( block of memory) in main memory that have recently been accessed